

EXHIBIT 10

The screenshot displays the Git Gud web interface. The browser's address bar shows the URL `https://gitgud.io/fiz1/oai-reverse-proxy/-/blob/main/README.md?ref_type=heads`. The interface includes a sidebar on the left with a 'Project' section containing a list of actions: Manage, Plan, Code, Build, Deploy, Operate, Monitor, and Analyze. The main content area is titled 'oai-reverse-proxy' and features a 'Features' section with a list of capabilities, including support for multiple APIs (OpenAI, Anthropic, AWS Bedrock, Google MakerSuite/Gemini API, Azure OpenAI), translation from OpenAI-formatted prompts, multiple API keys with rotation and rate limit handling, basic user management, simple role-based permissions, per-model token quotas, temporary user accounts, prompt and completion logging, and abuse detection and prevention. Below the features is a 'Usage Instructions' section, which includes a paragraph about running a self-hosted instance, a 'Self-hosting (locally or without Docker)' section with instructions on using services like ngrok or trycloudflare.com, a 'Self hosting (with Docker)' section with instructions on using Docker, and an 'Alternatives' section mentioning Fiz and Sekrit.

Git Gud Explore

Search or go to...

fiz123321 / oai-reverse-proxy

Project

- oai-reverse-proxy
- Manage
- Plan
- Code
- Build
- Deploy
- Operate
- Monitor
- Analyze

Features

- Support for multiple APIs
 - OpenAI
 - Anthropic
 - AWS Bedrock
 - Google MakerSuite/Gemini API
 - Azure OpenAI
- Translation from OpenAI-formatted prompts to any other API, including streaming responses
- Multiple API keys with rotation and rate limit handling
- Basic user management
 - Simple role-based permissions
 - Per-model token quotas
 - Temporary user accounts
- Prompt and completion logging
- Abuse detection and prevention

Usage Instructions

If you'd like to run your own instance of this server, you'll need to deploy it somewhere and configure it with your API keys. A few easy options are provided below, though you can also deploy it to any other service you'd like if you know what you're doing and the service supports Node.js.

Self-hosting (locally or without Docker)

Follow the "Local Development" instructions below to set up prerequisites and start the server. Then you can use a service like [ngrok](#) or [trycloudflare.com](#) to securely expose your server to the internet, or you can use a more traditional reverse proxy/WAF like [Cloudflare](#) or [Nginx](#).

Ensure you set the `TRUSTED_PROXIES` environment variable according to your deployment. Refer to [.env.example](#) and [config.ts](#) for more information.

Self hosting (with Docker)

If you have a Docker-capable VPS or server, use the Huggingface Dockerfile ([./docker/huggingface/Dockerfile](#)) to build an image and run it on your server.

Ensure you set the `TRUSTED_PROXIES` environment variable according to your deployment. Refer to [.env.example](#) and [config.ts](#) for more information.

Alternatives

Fiz and Sekrit are working on some alternative ways to deploy this conveniently. While I'm not directly involved in writing code or scripts for that project, I'm providing some advice and will include links to their work here when it's ready.

Help